深蓝学院
shenlanxueyuan.com

机器人中的数值优化

第一章作业分享

主讲人 坦克小白

纲要

- 问题描述
- 问题分析
- 代码实现

问题描述

深蓝学院
shenlanxueyuan.com

用线搜索方法求解 n 维 Rosenbrock 函数的最小值点:

$$f(x) = f(x_1, x_2, \cdots, x_n)$$

$$= \sum_{i=1}^{N/2} [100(x_{2i-1} - x_{2i})^2 + (x_{2i} - 1)^2]$$

$$= [100(x_1 - x_2)^2 + (x_2 - 1)^2] + \cdots$$

$$+ [100(x_{N-1} - x_N)^2 + (x_N - 1)^2]$$

问题分析

线搜索方法

- 线搜索方法即给定当前迭代点 x^k，以函数在当前梯度的负方向作为下降方向 $d^k =- \nabla f(x^k)$，按照一定的步长 α^k 确定下一个迭代点：

$$x^{k+1} = x^k + \alpha^k d^k$$

- 线搜索的步长 α^k 通过 Armijo 条件获得

问题分析

Armijo 条件

- 为了使函数充分下降，同时避免再费时费力的求解优化问题来获得最优的迭代步长，我们使用 Armijo 条件用较小的计算量获得一个次优的迭代步长：

$$f(x^k) - f(x^k - \alpha^k d^k) \geq - c \cdot \alpha_k \nabla f(x^k)^T d^k,\ c \in (0,1)$$

- 先定义一个初始步长 $\alpha^k = 1$，若初始步长不满足条件，则步长减半，如此往复，直至步长满足 Armijo 条件

代码实现

深蓝学院 shenlanxueyuan.com

Rosenbrock 函数的计算（二维）

- Rosenbrock 函数值的计算：用数组存储 (x_1, x_2)，再套公式计算

```
def Rosenbrock(x):
    return 100*(x[0]**2.0 - x[1])**2.0 + (x[0] - 1)**2
```

- Rosenbrock 梯度的计算：手动推导梯度表达式，再代入数值计算

```
def RosenbrockGradient(x):
    gradX1 = 400 * x[0] * (x[0]**2 - x[1]) + 2*(x[0] - 1)
    gradX2 = -200 * (x[0]**2 - x[1])
    grad = np.array([gradX1, gradX2])
    return grad
```

代码实现

深蓝学院
shenlanxueyuan.com

Rosenbrock 函数的计算（二维）

➢Armijo 条件：

算法 6.1 线搜索回退法

1. 选择初始步长 $\hat{\alpha}$，参数 $\gamma, c \in (0,1)$．初始化 $\alpha \leftarrow \hat{\alpha}$.
2. **while** $f(x^k + \alpha d^k) > f(x^k) + c\alpha \nabla f(x^k)^{\mathrm{T}} d^k$ **do**
3. 　令 $\alpha \leftarrow \gamma\alpha$.
4. **end while**
5. 输出 $\alpha_k = \alpha$.

```
def Armijo(x, grad):
    c = 0.1
    tau = 1
    x1 = x[0] - tau * grad[0]
    x2 = x[1] - tau * grad[1]
    nextX = np.array([x1, x2])

    while Rosenbrock(nextX) > Rosenbrock(x)
+ (c * tau) * np.dot(grad, grad):
        tau *= 0.5
        x1 = x[0] - tau * grad[0]
        x2 = x[1] - tau * grad[1]
        nextX = np.array([x1, x2])

    alpha = tau
    return alpha
```

代码实现

Rosenbrock 函数的计算（二维）

➤线搜索方法:

算法 1: Steepest Gradient Descent with Armijo

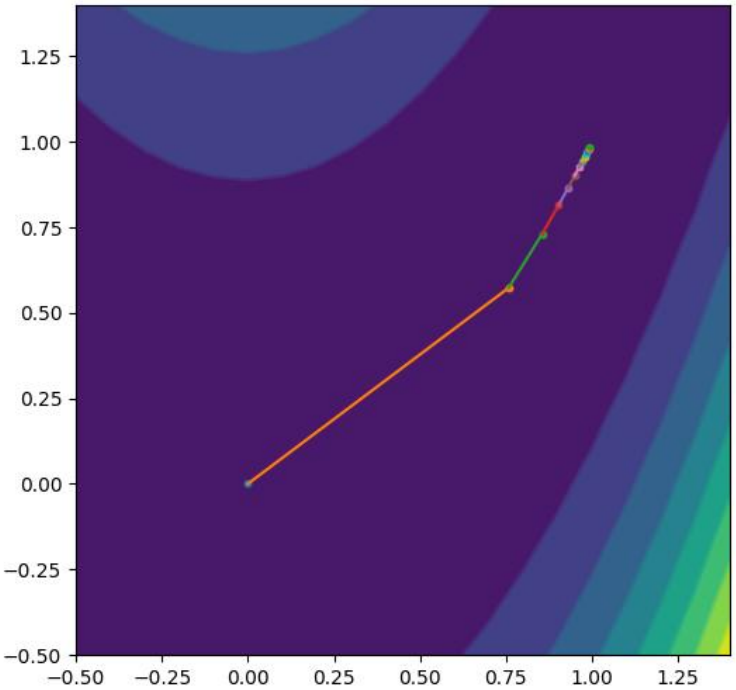
$k := 0, c := c_0, \mathbf{x} := \mathbf{x}_0, found := false$;
while *not found* **do**
  $\mathbf{g} := \nabla f(\mathbf{x})$;
  if $||\mathbf{g}|| < \epsilon$ **then**
    $found := true$;
  else
    $\alpha := 1, \mathbf{x}_{new} := \mathbf{x} - \alpha\mathbf{g}$;
    while $f(\mathbf{x}_{new}) > f(\mathbf{x}) - c\alpha\mathbf{g}^\top\mathbf{g}$ **do**
      $\alpha := \alpha/2, \mathbf{x}_{new} := \mathbf{x} - \alpha\mathbf{g}$;
    end
    $\mathbf{x} := \mathbf{x}_{new}, k := k + 1$;
  end
end

```
def LineSearch(x0):
    iter = 1
    maxIter = 5000
    x = x0
    error = 10
    tolerance = 0.01
    while (iter < maxIter) and (error >
tolerance):
        grad = RosenbrockGradient(x)
        error = np.linalg.norm(grad)
        alpha = Armijo(x, grad)
        X0 = x[0] - alpha * grad[0]
        X1 = x[1] - alpha * grad[1]
        x = np.array([X0, X1])
        iter += 1
    return x
```

代码实现

深蓝学院
shenlanxueyuan.com

Rosenbrock 函数的计算（二维）



代码实现

Rosenbrock 函数的计算（n 维）

- Rosenbrock 梯度的计算：找到表达式的规律再代数计算

```
    def RosenbrockGradient(self, x):
        x_middle = x[1:-1]
        x_head = x[:-2]
        x_tail = x[2:]

        gradient = np.zeros_like(x)
        gradient[0] = -400*x[0]*(x[1]-x[0]**2) - 2*(1-x[0])
        gradient[1:-1] = 200*(x_middle - x_head**2) - 400*(x_tail - 
x_middle**2)*x_middle - 2*(1-x_middle)
        gradient[-1] = 200*(x[-1]-x[-2]**2)

        return gradient
```

代码实现

深蓝学院 shenlanxueyuan.com

Rosenbrock 函数的计算（n 维）

- Armijo 条件：用 for 循环更新函数值

```
    def Armijo(self, x, grad):
        c = 0.02
        tau = 1.0
        alpha = tau
        x_next = np.zeros_like(x)
        for i in range(self.dimension):
            x_next[i] = x[i] - tau * grad[i]
        while self.Rosenbrock(x_next) > self.Rosenbrock(x) + (c * tau) *
np.dot(grad.T, grad):
            tau = tau * 0.4
            for i in range(self.dimension):
                x_next[i] = x[i] - tau * grad[i]
            alpha = tau
        return alpha
```

代码实现

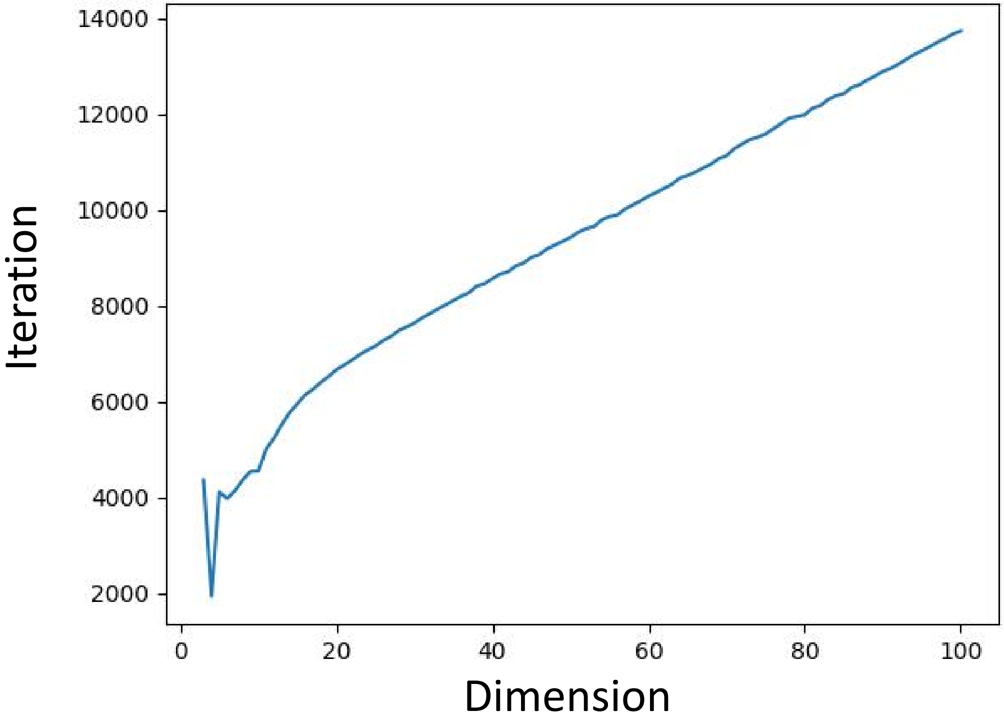
Initial Conditions:

- Tolerance $= 0.001$
- Armijo:

  $c = 0.02$

  $\alpha_0 = 1$

维度	迭代次数	精度
2	4862	0.000998
3	4357	0.000982
4	1939	0.000995
…	…	…
100	13733	0.0099



在线问答

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening